
Dozent: Prof. Dr. Stefan Weinzierl
Tutor: Athanasios Lykartsis

MATLAB Aufgabe 3 (Gesamtpunktzahl: 20)

Bei der dritten Aufgabe geht es um die Themen *Diskrete Fourier-Transformation*, *Fensterung*, *Abtastung*, *Aliasing* und *DFT-Synthese*. Als Soundmaterial werden mehrere Instrumentenaufnahmen (Gitarre, Oboe, Saxofon usw.) bereitgestellt werden, aus denen Sie zwei auswählen können. Die Sounddateien haben eine Abtastrate von $f_s = 44100$ Hz sind in Stereo-Format, insofern können Sie nur den einen Kanal behalten.

- 1.** Programmieren Sie eine Implementierung der diskreten Fourier-Transformation in MATLAB. Erstellen Sie eine Funktion, die als Eingabe einen Vektor beliebiger Länge N und eine Abtastrate f_s bekommt, und als Ausgabe die komplexe diskrete-Transformierte, das Betrags- und Phasenspektrum und eine Grafik mit einem plot für das Betragsspektrum und einem für das Phasenspektrum von 0 bis $f_s/2$. Die Funktion kann die Form $[X \text{ abs}X \text{ angle}X] = \text{dft_imp}(x, f_s)$ haben. Sie können für die tatsächliche Berechnung eine FOR-Schleife benutzen, um die Summenbildung der DFT zu realisieren. Achten Sie dabei auf die Symmetrie der diskreten Fourier-Transformation, den Vorfaktor zur Skalierung des Spektrums und die richtige Achsenskalierung. Erstellen Sie am Ende eine neue Funktion durch geeignete Modifizierung der Ersten, die eine IDFT implementiert. **(5 Punkte)**
- 2.** Benutzen Sie die von Ihnen in **1.** erstellte Funktion, um die zwei Sounddateien zu transformieren (nachdem Sie die erstmal in MATLAB importiert haben) und vergleichen Sie die zwei Betragsspektren. Erstellen Sie anschließend einen Sinusverlauf von 1 kHz mit einer Länge von 2 Sekunden und für eine Abtastrate von 8 kHz. Stellen Sie die Grundperiode der Signale in Samples fest. Benutzen Sie die MATLAB Fensterfunktionen *rect()*, *hann()*, *bartlett()* um jeweils zwei Fenstervarianten zu erzeugen: Eines, was ein ganzzahliges Vielfaches (zwischen 10 und 20) der Grundperiode als Länge hat, und eines, was die gleiche Länge plus ein Viertelperiode hat. Fenstern sie jeweils ein Sinussignal mit den zwei Fenstervarianten, transformieren Sie das Ergebnis und vergleichen Sie die Betragsspektren. **(5 Punkte)**
- 3.** Erzeugen Sie noch zwei Sinussignale mit den Frequenzen 3 und 6 kHz, fenstern Sie mit den davor erstellten Fenstern und benutzen Sie die MATLAB *fft()* Funktion um zu transformieren und die Betragsspektren zu bekommen. Was merken Sie für die Betragsspektren im Vergleich zu denen aus Aufgabenteil **2**? Benutzen Sie das mit der *rect()* Funktion gefensterte Sinussignal von 3 kHz (beide Varianten) und erweitern Sie den Vektor mit genauso vielen Nullen wie die Länge des Fensters (Zero-Padding). Transformieren Sie erneut und diskutieren Sie die Ergebnisse hinsichtlich der Effekte des Zero-Paddings. **(5 Punkte)**
- 4.** Erzeugen Sie ein Betrags- und ein Phasenspektrum, sodass mithilfe der DFT-Rücktransformation (*ifft()* Funktion in MATLAB) ein reelles Sinussignal der Frequenz 400 Hz und eine Länge von **exakt** 1000 Samples entstehen kann (die Abtastrate f_s bleibt bei 8 kHz).

Dazu benutzen Sie die Symmetrie der DFT und achten Sie noch darauf, dass aus den zwei Spektren **ein gesamtes komplexes Spektrum** erzeugt werden soll, was dann als Eingabe für die inverse DFT dient. Überprüfen Sie die Ergebnisse, indem Sie das Signal plotten und sich anhören. **Bonus:** Experimentieren Sie indem Sie ein komplexeres Spektrum (z. B. das eines Sinus-Sweeps) erzeugen und das zugehörige Zeitsignal herstellen. **(5 + 3 Punkte)**

Abgabetermin ist der **Freitag, 15. Februar 2013** (am Ende des Tages). Abgegeben werden müssen der Code (.m Dateien), alle generierten Grafiken und eine .pdf Datei mit einer kurzen Diskussion der Ergebnisse, in einer .zip Datei mit den Namen der Gruppenmitglieder (z. B. name1_name2_name3.zip). Die benötigten Funktionen für die Durchführung der Aufgabe sind: *wavread(), plot(), semilogx(), subplot() axis(), input(), sin(), length(), size(), linspace(), zeros(), ones(), sound(), soundsc(), max(), abs(), angle(), filter(), rect(), hann(), bartlett(), fft(), ifft(), exp()*.

Viel Erfolg!